Multi-Label Classification with Conditional Tree-structured Bayesian Networks

Original work: Batal, I., Hong C., and Hauskrecht, M. An Efficient Probabilistic Framework for Multi-Dimensional Classification. CIKM 2013.

Charmgil Hong

Department of Computer Science
University of Pittsburgh
Machine Learning
Motivation

• Traditional classification
  • Each data instance is associated with a single class variable
Motivation

• Multi-label classification
  • In many real-world applications, each data instance can be associated with \textit{multiple class variables}
  • Examples
Motivation

- Multi-label classification
  - In many real-world applications, each data instance can be associated with multiple class variables

- Examples
  - A news article may cover multiple topics such as politics and economics
Motivation

• Multi-label classification
  • In many real-world applications, each data instance can be associated with **multiple class variables**

• Examples
  • An image may include multiple objects as *beach, building* and *bridge*

Taken from the Scene dataset [Boutell et al., '04]
Motivation

• Multi-label classification
  • In many real-world applications, each data instance can be associated with multiple class variables
• Examples
  • A gene may be associated with several biological functions
Motivation

• Multi-label classification
  • Each data instance is associated with multiple class variables
  • Objective: assign to each instance the most probable assignment of the class variables

Class 1 ∈ \{ red, blue \}
Class 2 ∈ \{ ◦, △, □ \}
Motivation

• Simplest solution
  • Learning $d$ independent classifiers for $d$ class labels
  • It does not capture the dependency relations between the classes
CTBN

- Conditional Tree-structured Bayesian Network
CTBN

• Conditional Tree-structured Bayesian Network (CTBN) for modeling $P(Y_1, \ldots, Y_d|X)$

  • A class variable can have at most one other class variable as a parent (the dependencies among classes form a directed tree)

  • The feature vector $X$ is the common parent for all class variables

An example CTBN
CTBN

- Conditional Tree-structured Bayesian Network (CTBN) for modeling $P(Y_1, ..., Y_d|X)$
  - A class variable can have at most one other class variable as a parent (the dependencies among classes form a directed tree)
  - The feature vector $X$ is the common parent for all class variables

An example CTBN
CTBN

• Conditional Tree-structured Bayesian Network (CTBN) for modeling \( P(Y_1, \ldots, Y_d | X) \)
  • A class variable can have at most one other class variable as a parent (the dependencies among classes form a directed tree)
  • The feature vector \( X \) is the common parent for all class variables

An example CTBN
CTBN

- Conditional Tree-structured Bayesian Network (CTBN) for modeling $P(Y_1, \ldots, Y_d|X)$
  - A class variable can have at most one other class variable as a parent (the dependencies among classes form a directed tree)
  - The feature vector $X$ is the common parent for all class variables
- We restrict the dependency structure to a tree because:
  1. The optimal structure can be learned efficiently (coming up)
  2. Exact inference can be done in $O(d)$ time (not in this talk)
Representation

• The conditional class distribution is:

\[ P(y_1, \ldots, y_d | \mathbf{x}) = \prod_{i=1}^{d} P(y_i | \mathbf{x}, y_{\pi(i,T)}) \]

• It is the product of the dependencies in the network
Representation

- The conditional class distribution is:
  \[ P(y_1, \ldots, y_d | x) = \prod_{i=1}^{d} P(y_i | x, y_{\pi(i,T)}) \]

- It is the product of the dependencies in the network
Representation

- The conditional class distribution is:
  \[ P(y_1, \ldots, y_d | x) = \prod_{i=1}^{d} P(y_i | x, y_{\pi(i,T)}) \]
  
- It is the product of the dependencies in the network

An example CTBN
Representation

- The conditional class distribution is:
  \[ P(y_1, \ldots, y_d|x) = \prod_{i=1}^{d} P(y_i|x, y_{\pi(i,T)}) \]

- It is the product of the dependencies in the network

An example CTBN
Representation

• The conditional class distribution is:

\[ P(y_1, \ldots, y_d | x) = \prod_{i=1}^{d} P(y_i | x, y_{\pi(i,T)}) \]

• It is the product of the dependencies in the network

This network represents

\[ P(y_1, y_2, y_3, y_4 | x) = P(y_3 | x) \cdot P(y_2 | x, y_3) \cdot P(y_1 | x, y_2) \cdot P(y_4 | x, y_2) \]
Representation

- The conditional class distribution is:

\[ P(y_1, \ldots, y_d | x) = \prod_{i=1}^{d} P(y_i | x, y_{\pi(i,T)}) \]

- It is the product of the dependencies in the network

- Each \( P(y_i | x, y_{\pi(i,T)}) \) is represented by classifier functions.

This network represents

\[ P(y_1, y_2, y_3, y_4 | x) = P(y_3 | x) \cdot P(y_2 | x, y_3) \cdot P(y_1 | x, y_2) \cdot P(y_4 | x, y_2) \]
Representation

- The conditional class distribution is:

\[ P(y_1, \ldots, y_d | x) = \prod_{i=1}^{d} P(y_i | x, y_{\pi(i,T)}) \]

- It is the product of the dependencies in the network

- Each \( P(y_i | x, y_{\pi(i,T)}) \) is represented by classifier functions.

\[ f_{1,0}(X) = P(Y_1 | X, Y_2 = 0) \]
\[ f_{1,1}(X) = P(Y_1 | X, Y_2 = 1) \]
Structure learning

- **Objective:** Find the tree structure that best approximates $P(Y|X)$, i.e., that maximizes the conditional log-likelihood of data

- **Idea:** Cast the structure learning into the maximum branching tree problem [Edmonds, ’67]

- **Next:** Illustration through the example CTBN
Structure learning

1. Define a complete weighted directed graph $G$
Structure learning

1. Define a **complete weighted directed graph** $G$
   - Draw $d$ nodes for all class variables $Y_i: i \in \{1, \ldots, d\}$
Structure learning

1. Define a complete weighted directed graph $G$
   - Draw $d$ nodes for all class variables $Y_i: i \in \{1, \ldots, d\}$
   - Connect all the node pairs and add self-loops with directed edges
Structure learning

2. Compute the edge weights using \textit{conditional log-likelihood} of the data:

\[
W_{j \rightarrow i} = \sum_{(x^{(k)}, y^{(k)}) \in D} \log P(y_i^{(k)} | x^{(k)}, y_j^{(k)})
\]
Structure learning

3. Find the tree that maximizes the sum of the edge weights by solving the maximum branching tree problem
Structure learning

4. The corresponding CTBN model is ...
Experiments

• Compared methods
  • *Binary Relevance (BR)* [Boutell et al., ’04, Clare et al., ’01]
  • *Classification with heterogeneous features (CHF)* [Godbole and Sarawagi, ’04]
  • *Multi-label k-nearest neighbor (MLKNN)* [Zhang and Zhou, ’07]
  • *Instance-based learning by logistic regression (IBLR)* [Cheng and Hüllermeier, ’09]
  • *Classifier chains (CC)* [Read et al., ’09]
  • *Maximum margin output coding (MMOC)* [Zhang and Schneider, ’12]
Experiments

• Data
  • 10 publicly available datasets from different domains

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Instances</th>
<th># Features</th>
<th># Classes</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emotions</td>
<td>593</td>
<td>72</td>
<td>6</td>
<td>Music</td>
</tr>
<tr>
<td>Yeast</td>
<td>2,417</td>
<td>103</td>
<td>14</td>
<td>Biology</td>
</tr>
<tr>
<td>Scene</td>
<td>2,407</td>
<td>294</td>
<td>6</td>
<td>Image</td>
</tr>
<tr>
<td>Enron</td>
<td>1,702</td>
<td>1,001</td>
<td>53</td>
<td>Text</td>
</tr>
<tr>
<td>TMC 2007</td>
<td>28,596</td>
<td>30,438</td>
<td>22</td>
<td>Text</td>
</tr>
<tr>
<td>RCV1_subset1</td>
<td>6,000</td>
<td>8,394</td>
<td>10</td>
<td>Text</td>
</tr>
<tr>
<td>RCV1_subset2</td>
<td>6,000</td>
<td>8,304</td>
<td>10</td>
<td>Text</td>
</tr>
<tr>
<td>RCV1_subset3</td>
<td>6,000</td>
<td>8,328</td>
<td>10</td>
<td>Text</td>
</tr>
<tr>
<td>RCV1_subset4</td>
<td>6,000</td>
<td>8,332</td>
<td>10</td>
<td>Text</td>
</tr>
<tr>
<td>RCV1_subset5</td>
<td>6,000</td>
<td>8,367</td>
<td>10</td>
<td>Text</td>
</tr>
</tbody>
</table>
Experiment Results

- **Exact Match Accuracy**
  
  *The probability of all classes being predicted correctly (higher is better)*
## Experiment Results

- **Exact Match Accuracy**

  *The probability of all classes being predicted correctly (higher is better)*

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BR</th>
<th>CHF</th>
<th>MLKNN</th>
<th>IBLR</th>
<th>CC</th>
<th>MMOC</th>
<th>CTBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emotions</td>
<td>0.266</td>
<td>0.315</td>
<td>0.283</td>
<td>0.332</td>
<td>0.272</td>
<td>0.336</td>
<td>0.335</td>
</tr>
<tr>
<td>Yeast</td>
<td>0.147</td>
<td>0.162</td>
<td>0.179</td>
<td>0.204</td>
<td>0.194</td>
<td>0.214</td>
<td>0.195</td>
</tr>
<tr>
<td>Scene</td>
<td>0.521</td>
<td>0.160</td>
<td>0.629</td>
<td>0.644</td>
<td>0.633</td>
<td>0.684</td>
<td>0.626</td>
</tr>
<tr>
<td>Enron</td>
<td>0.162</td>
<td>0.169</td>
<td>0.078</td>
<td>0.163</td>
<td>0.173</td>
<td>-</td>
<td>0.168</td>
</tr>
<tr>
<td>TMC</td>
<td>0.315</td>
<td>0.322</td>
<td>0.165</td>
<td>0.316</td>
<td>0.323</td>
<td>-</td>
<td>0.329</td>
</tr>
<tr>
<td>RCV1_subset1</td>
<td>0.278</td>
<td>0.357</td>
<td>0.205</td>
<td>0.279</td>
<td>0.429</td>
<td>Could</td>
<td>0.448</td>
</tr>
<tr>
<td>RCV1_subset2</td>
<td>0.42</td>
<td>0.466</td>
<td>0.288</td>
<td>0.417</td>
<td>0.517</td>
<td>not</td>
<td>0.531</td>
</tr>
<tr>
<td>RCV1_subset3</td>
<td>0.442</td>
<td>0.485</td>
<td>0.327</td>
<td>0.446</td>
<td>0.54</td>
<td>finish</td>
<td>0.561</td>
</tr>
<tr>
<td>RCV1_subset4</td>
<td>0.494</td>
<td>0.532</td>
<td>0.354</td>
<td>0.491</td>
<td>0.579</td>
<td>-</td>
<td>0.59</td>
</tr>
<tr>
<td>RCV1_subset5</td>
<td>0.412</td>
<td>0.457</td>
<td>0.276</td>
<td>0.411</td>
<td>0.497</td>
<td>-</td>
<td>0.538</td>
</tr>
</tbody>
</table>

**# win-tie-loss**: 9-1-0 8-2-0 7-3-0 9-1-0 6-4-0 0-1-2
Experiment Results

- **Normalized conditional log-likelihood loss**
  
  *Negative log-likelihood normalized on each dataset (lower is better)*
Experiment Results

• **Normalized conditional log-likelihood loss**

  Negative log-likelihood normalized on each dataset (lower is better)
Conclusion

• We proposed a novel probabilistic approach to multi-label classification
  • CTBN encodes the conditional dependence relations between classes
  • Efficient structure learning and exact inference algorithms are presented
  • Our approach outperforms several state-of-the-art methods
Thank you!